BACK TO BASICS AGAIN -- A SCIENTIFIC DEFINITION OF SYSTEMS ENGINEERING

Brian W. Mar
University of Washington
Box 352700
Seattle, WA 98915

ABSTRACT

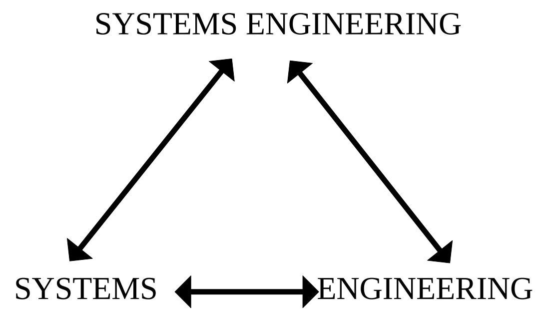
The major confusion with the understanding of systems engineering and improving its scientific basis is the failure to define the system of interest. Given a clear definition of the system of interest, engineered functions of that system can be identified, and applications of systems engineering concepts to those activities can be examined.

A systems framework is suggested to classify basic engineering and systems engineering activities. This framework allows a better definition and search for scientific foundations of systems engineering.

INTRODUCTION

Since the beginning of INCOSE (formerly NCOSE), I have emphasized the need to understand the basic fundamental scientific concepts for all systems engineering activities. These basic concepts are needed to focus the continuing discussion of (1) what is systems engineering, (2) whether it is a process or skill code, and (3) what is the role of systems engineering in the engineering of complex systems.

This paper begins with a review of the basic concepts from engineering, general systems theory, and design science that can help define a science for systems engineering. Given this background a set of laws for systems engineering are proposed and this framework is examined.



Systems engineering incorporates the elements of engineering with some basic systems concepts. This often leads to the confusion that systems engineering is a subset of engineering. I suggest that systems engineering is a superset of engineering that applies systems concepts.

The major confusion with the understanding of systems engineering is the failure to define the system of interest. Given clear definition of a system of interest, functions of an engineered system can be identified, and applications of systems engineering concepts to those activities can be debated.

ENGINEERING CONCEPTS

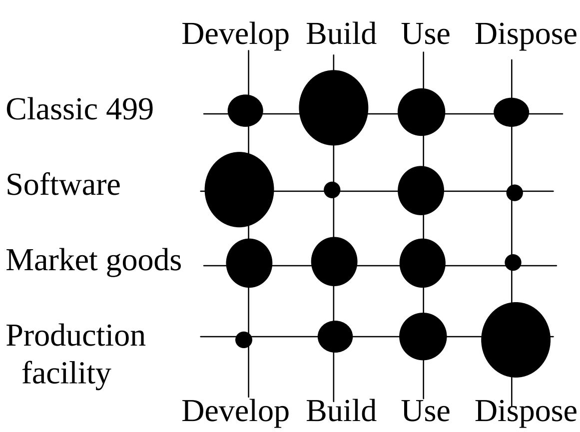
The usual definition of engineering in most dictionaries suggests ***activities*** such as "putting scientific knowledge to practical uses” or “planning, designing, construction or management of products of technology.” Harwell, Lake, Martin, and Velman (1996) present an anatomy of the engineering of systems as subsystems of the engineering process such as:

Development
Production
Test
Deployment
Training
Support
Disposition

Each of these activities generates, in sequence or in parallel, more detailed views of the end-product and other products needed to create that end-product. These activities are often labeled as lifecycle activities in a product development effort. The actual allocation of resources to these activities is a function of the life cycle model selected and the type of system of interest. Harwell et al (1996) subsystems can be lumped into 4 groups:

Develop
Build
Use (test, deploy, train, support)
Dispose

The relative magnitide of life cycle costs components for different types of products are shown below:



A classic implementation of the systems engineering paradigm suggests that it is cost effective to define requirements before designing the product. This may not hold true when initial decisions are not major drivers of the total lifecycle costs or when the profit to development cost ratio is high..

There is a growing literature on the science of design and design science in the lifecycle context (Hubka and Eder, 1996). Mar (1996) reviewed these literature at the Boston INCOSE meeting and mapped efforts to make design a science and to make systems engineering a science. Both the design science and systems engineering science groups are seeking to understand and improve the lifecycle process of engineering.

SYSTEMS CONCEPTS

Basic systems concepts traced back to the 1800s are:

- The whole is more than the sum of its parts
- The whole determines the nature of the parts
- The parts cannot be understood if considered in isolation from the whole
- The parts are dynamically interre lated or interdependent.

Systems can be classified by the following characteristics:

living or non living
abstract or concrete
open or closed
high or low degree of entropy
level of organization of complexity
function(s) of system
degree of feedback

General systems theory started in the 1950s and attempted to develop a third branch of science at par with mathematics and philosophy. Von Neumann (1948) helped lay the foundations of Artificial intelligence, Shannon (1949) defined communication theory, Wiener (1948) stimulated cybernetics, and Ashby (1956) stimulated information and control theory. These efforts lead the search for a unity of science to apply a pattern of scientific theory with different degrees of complexity to hierarchical levels.

Some trace the systems approach back to the Radio Corporation of America in 1930s. Churchman (1968) described the systems approach as a process or common framework that was an application of general systems theory or a new kind of scientific method (van Gigch, 1974). The systems approach is the basis for systems engineering. Hall (1962) traced the beginnings of systems engineering to G.W. Gilman, Director of Systems Engineering at Bell Telephone Laboratories.

The systems approach treats problems as systems to be described, developed, maintained or altered. Simply stated the systems approach advocates:

1. Understand the problem before attempting to solve it as a system

2. Identify and rank all possible solutions prior to selecting an answer

3. Looking for hybrid solutions to add to the set of alternatives

4. Select a solution, capture the analysis and formulate the subsequent problem or implementation of the solution.

Defining anything as a system is a basic concept essential to systems engineering. Thus, it is important as the first step in any systems engineering effort to define the system and the abstraction used to organize the parts of that system. The sequential decomposition of the whole system into its parts (viewing each part as another system) leads to the hierarchical decomposition used as the framework for most systems engineering efforts. While hierarchical frameworks indicate decomposition, flow diagrams are needed to indicate the complex interactions of these components.

Once the system is identified, it must be classified to determine if the system of interest is a fabricated technological end-product that is created using the engineering process, or whether the system itself is a process that may or may not be engineered. For example the continual debate on whether systems engineering has a management component, is really addressing whether or not system such as organizations, or life cycle activities should be systems engineered or whether organization or management theory should be applied.

Basic systems concepts used in systems engineering are (1) identifying system boundaries, (2) identifying inputs and outputs to a system, and (3) understanding the functions that the systems perform to convert inputs to outputs.

DESIGN SCIENCE CONCEPTS

Warfield (1994) has formulated six laws of generic design that may provide a point of departure for developing a science of systems engineering. Two of these laws are generic and apply to any scientific issue rather than just to design

1. **Law of Limits** is a restatement of systems concept that a system is bounded and has inputs and outputs

2. **Law of Gradation** is another restatement of system concepts that allows the division of a whole into its parts. Repeating this process creates a hierarchy of views ranging from general top level descriptions to highly detailed and specific lower level descriptions.

The four Warfield laws that are design specific are:

1. **Law of Requisite Variety** suggests that you need as many answers as you have problems. If you have fewer your system is undefined and if you have more your system is overdefined. This can apply to requirements/answer pairs.

2. **Law of Requisite Parsimony** suggests that you should not overload the problem solver with information at a rate they can not process. The rule of seven by Miller is another version of this law.

3. **Law of Requisite Saliency** suggests that those involved in a development effort have radically different views of the problem and solution space, and some process must be introduced to develop a shared vision among theses participants.

4. **Law of Success and Failure** for Generic Design defines seven factors for success: leadership, financial support, component availability, design environment, designer participation, document support, and design process. Current efforts to establish a systems engineering capability model include these factors.

SYSTEMS ENGINEERING BASICS

The basic concepts for systems engineering that I have identified are:

1. VIEW ANYTHING AS A SYSTEM
View anything to be systems engineered as a system -- define its boundaries, its inputs and outputs, and basic systems characteristics (at least determine if it is an end-product or a ***process*** to develop an end-product)

2. SYSTEM DESCRIPTIONS EXPAND AS THE LIFECYLE PROCEEDS
For either a product system or a process system, the systems engineering process will develop increasingly detailed descriptions of the system of interest.

3. FOUR VIEWS OF ANY SYSTEM MUST BE GENERATED
In order to provide a shared vision of these systems, at least four views of any system must be generated at each increasing detailed level of description:

FUNCTIONAL VIEW - describing the behavior of the system
REQUIREMENTS VIEW - describing how well the functions must be performed by the system
ANSWER VIEW - describing what performs the functions (hardware, software, peopleware, etc.)
TEST VIEW - describing how to evaluate if the answer performs the functions as required and the results of such evaluation

4. COMPLETE ALL FOUR VIEWS BEFORE DECOMPOSING ANY VIEW
All four views must be generated before the next step in the life cycle of engineering begins. A shared vision developed by all par-

ties interested in the acquisition, development and use of the product must be captured in these four views.

5. NO SINGLE LIFECYCLE MODEL
Different end-products may require different processes to effectively create them. The end-product descriptions are used to classify the product type and select the most appropriate process system to develop that end product.

6. MANAGEMENT/PROCESS SYSTEMS
The process system is described by functions and requirements that specify how the product system descriptions are generated, managed and used to create the end-product. Thr answer and test views of the process system defines tasks and activities that must be performed to develop and control the process infrastructure and support systems needed to produce the end product system.

7. DOCUMENT BASED DESCRIPTIONS ARE OBSOLETE
Development and use of these data in textual format will be ambiguous and static and introduce high program risks. Unless these data are captured in dynamic data bases with simulation capability, these risk cannot be managed. Control of the product descriptions or process system performance is very difficult without such capability.

8. PEOPLE SYSTEMS INTRODUCE MAJOR PROGRAM RISKS
While nonliving systems that are engineered may be fully defined, living systems including those with people may never be completely understood. The practice of viewing people as robots that can be trained or selected to perform desired tasks is not without risk. Risk assessment and management are necessary to minimize major failures when engineering living systems.

APPLICATION OF BASIC CONCEPTS

These basic systems engineering concepts provides a framework to develop a science that may help improve the shared vision and clarify discussions of different views of systems engineering. Many of the current debates on the nature and goals of systems engineering can be traced to misunderstandings over the actual system of interest.

For example in the definition of engineering by Harwell et al. (1996), the engineering system defined subsystems that developed an end-product, as well as the description of the end-product itself. By clearly identifying which major subsystems describe the end product characteristics and which subsystems describe the development and operation systems, the reader may appreciate the difference between these two systems. Presenting four views of each of these systems provides a richer context for discussion of the first 7 concepts presented.

Another example of how the suggested concepts may help the discussion of systems engineering ideas is Sheard's (1996) description of 12 types of systems engineering activities. A greater insight to the nature of systems engineering may be realized when the roles identified by Sheard are classified into two subsystems. Six of these roles are activities that implement the lifecycle activities identified by Harwell, et al. (1996). Table 1 compares the roles and life cycle activities and suggests that Sheard is mapping these roles with life cycle tasks.

Traditionally, individuals labeling themselves systems engineers focused on a particular lifecycle activity, but not all of them. This leads to the current debate over what systems engineers do. If systems engineering is a super set of engineering, then it should have roles in all of the other lifecycle activities.

Table 1. SE Roles and Activities (Sheard, 1996, and Harwell et al, 1996)

Sheard	Harwell
Customer interface	Development
System designer	
System Analyst	
Requirements owner	
	Production
Validation and Verification	Test
Logistics and operations	Deployment
	Training
	Support
	Disposition

Sheard (1996) presents another important concept where six other roles are identified for systems engineering, five of which are more generic in nature and can be applied to any life cycle activities. These are

Process Engineer
Technical Manager
Glue Among Subsystems
Coordinator of Disciplines
Information Manager

These roles are communication, orchestration, and discipline types of roles. They are the basic systems engineering roles that can be applied to any engineering activity and are the real strength of systems engineering that complements domain knowledge, and system perspectives.

The last role identified by Sheard as Classified Adds SE is the use of the word systems engineer to identify experts for specific hardware and software systems. Example of this use is job titles such as software systems engineer, telephone systems engineers, and in the extreme solid waste systems engineer. This role identifies specific product domain rather than lifecycle or orchestration knowledge.

Roles such as customer interface, requirements owner, systems analyst and even system designers can be traced to the traditional DoD definitions of systems engineering where the tasks were to capture and translate customer needs, develop a comprehensive set of requirements, select and refine an answer, and then oversee the development and testing of the end-product. Commercial activities may employ different life cycle models but the communication, orchestration, and discipline roles for systems engineering remain.

The current confusion over what is systems engineering, and what is a systems engineer can be resolved by first identifying what type of system is being addressed, what are the characteristics of set of life cycle activities selected for the process system, and what are the characteristics of the end-product systems. I have described the needs to clearly identify the systems of interest prior to identifying how, where and when systems engineering can be of value (Mar, 1996b).

Not all end-product systems require the classical systems engineering process, it depends upon the size of the production run, the type of customer (market versus single customer), the ratio of development to production cost, the ratio of facility products to end-product, the number of subcontractors or suppliers, etc. (Mar, 1996b) Research is required to establish which life cycle models are more appropriate for each type of end-product system.

The classic concept of attempting to identify all requirements prior to the initiating of design activities is based on the assumption that (1) there are experts that know what all the requirements are and can effectively implement the systems engineering effort needed to collect them and organize them, (2) the costs of changing the selected answer to respond to additional requirements is very high, and (3) there is a process that allows control of the development process to ensure that all requirements are understood and addressed.

Systems engineering basics can be applied to reverse systems engineering as well as the classic processes. The selection of what type of life cycle is a key systems engineering role, and it in turn defines the type of life cycle activities that will benefit from systems engineering concepts.

If the end-product system and the process system selected to create the end-product do not satisfy these assumptions, the conventional systems engineering model of do it right the first time by defining requirements prior to seeking solutions may not be appropriate. In the case of software development, the cost of production is trivial while the cost of engineering (including code development) is very high. This is just the opposite of complex hardware system development. Mistakes in software engineering do not create major production costs, and many software processes use the strategy "build a little, test a little" to develop functions and requirements for their product.

There are many consumer goods that are created to provide unusual features that are used to gain market shares using advertising, rather than determining customer needs or desires. When a function/requirement cannot be met by the announced market date, the product may still be marketed and the added functions incorporated into the next model of that product. There is very little loss associated with not meeting requirements in such cases since being first to the market place defines market share.

Systems engineering can be used to select the most appropriate life cycle strategy, to develop a process to implement that life cycle strategy, and to capture the design information describing the end-product. These functions and their requirements need to be clearly defined. The answers may not be the traditional systems engineering answers or roles, but they will respond to the generic problem definition and problem solving concepts identified in this paper.

I suggest the function of systems engineering is to bring structure and discipline to the engineering process, and to provide the glue, the communication, and the direction for the reduction of the chaos associated with many engineering efforts. It is these activities that must be captured in the science of systems engineering.

CONCLUSIONS

INCOSE needs to walk the talk, we need to apply systems concepts and the systems approach to our definition of systems engineering and systems engineering science. We need to clearly define the types of product and process systems of interest and not try to apply a prescription for addressing one type of system on other types of systems where the prescription may be inappropriate. We need to identify generic concepts that improve creating shared visions, clearly and completely defining systems at many levels of resolution using the four basic views of systems, and we need to provide systems analysis abilities to evaluate alternative process and product solutions, to ensure that the right answer is selected for the right problem, rather than the right answer for the wrong problem.

REFERENCES

Ashby, R.W., 1956, Introduction to Cybernetics, Chapman and Hall, London

Churchman, C.W., 1968, The Systems Approach, Delacorte, New York

Hall,A.D, 1962, A Methodology for Systems Engineering, D. van Nostrand Co., New York

Harwell, R., J.G. Lake, J. N. Martin, J. Velman (1996) Anatomy of the Engineering of a System, Sixth Annual International INCOSE Symposium, Boston MA, p 489-494

Hubka, V., and W.E. Eder, 1996, Design Science--Introduction to the Needs, Scope and Organization of Engineering Design Knowledge, Springer, London

Mar, B.W., 1994, Systems Engineering Basics, Journal of NCOSE, Vol. 1, No. 1, p17-28

Mar, B.W., 1996, Improving the Design Component of Engineering Education, 6th Annual INCOSE Symposium, Boston, MA

Mar, B.W. 1996b, Whats in a Name, Systems Engineering in the Commercial World, Region II, INCOSE Miniconference, San Diego, Nov, 1996

Miller, G.A., (1956)The Magical Number Seven, Plus or Minus Two, Psychology Review 63 (2), 81-97

Shannon, C.E. and W. Weaver, 1949, The Mathematical Theory of Communication, University of Illinois Press

Sheard, S, 1996, Twelve Systems Engineering Roles, 6th Annual INCOSE Symposium, Boston, MA

van Gigch, P. J., 1974, Applied General Systems Theory, Harper & Row, New York

von Neumann, J., 1951, The General and Logical Theory of Automata, In L.A. Jeffress (ed) Cerebral Mechanisms in Behavior, Wiley, New York

Warfield, J. N., 1994, A Science of Generic Design, Iowa State University Press, Ames,

Wiener, N., 1948, Cybernetics, Wiley, New York

AUTHOR BIOGRAPHY

Brian W. Mar is Professor of Civil and Systems Engineering at the University of Washington in Seattle. He holds the Boeing Sutter Professorship in systems engineering.

He was one of the original organizers of INCOSE, and served as president in 1993. He has contributed to the development of the basic concepts of systems engineering and has creating teaching materials for undergraduate, graduate and continuing systems engineering education.

ACKNOWLEDGMENTS

The suggestions and comments of Jerry Lake Barney Morais and Sarah Sheard are greatly appreciated. The concepts and analysis presented represent my views and not those of the reviewers.